

**REMARKS****Status of this application**

In the Office Action mailed January 30, 2004, claims 1-20 were rejected under 35 U.S.C. 103(a) as being unpatentable over Chang et al. ("Chang"), U.S. Patent No. 6,584,459, in view of Chau et al. ("Chau"), U.S. Patent Application Publication No. US 2002/0123993 filed on Nov. 29, 2000 claiming priority based on U.S. Provisional Application No. 60/168,659 filed on December 2, 1999.

This response amends the specification to correct minor errors and requests reconsideration of the obviousness rejection for the reasons presented below:

**Independent claim 1 and its dependent claims 2-13**

Chang discloses a method for storing an XML document internally within a table column as a single CLOB (character based large object). An "XML extender" provides a new abstract data type called DB2XML that enables SQL operations to be performed on the XML object (Chang, col. 3, lines 53-60).

Contrary to the Examiner's suggestion, Chang does not disclose "*parsing the character data in said XML document to identify characters representing data values within at least some of the elements of said XML document [and then] storing each of said data values in a specified column location in one or more specified rows of one or more specified tables in said relational database system*" as claimed.

Chang describes a system in which the entire XML document is stored at in a single column (of type DB2XML) in a table. The XML parser 160 in Fig. 2 is used to determine whether a DTD is imbedded in the XML document as explained at col. 10, lines 10-19, to process the DTD as described at col. 14, lines 44-47, to count the tags in an XML document as described at col. 16, lines 56-59, and to parse XML documents when queries are performed on their contents. None of the parsing operations cited by the Examiner are employed to identify data values within XML elements which are then stored in specified column locations in a table as claimed.

The cited passage in Chang at col. 3, lines 26-64 describes nothing other that the storage of entire XML documents in single column locations. The passage at col. 12, lines 36-48 describes storing information which identifies and describes a DTD in a table row and does not suggest the

storage of element data in column locations. The passage at col. 13, lines 28+ describes creating XML tables which store entire XML documents in column locations rather than storing element data values from the XML document in column locations as claimed. Finally, the cited passage at col. 26, lines 9-34 refers to some of the claims – and it is guessed that the Examiner meant to specify col. 25, lines 9-34 which describes the use of the XML extender operators to perform SQL queries on entire XML documents stored in the DB2XML columns of a table. None of these cited passages discloses or suggests parsing an XML document to extract and store element data in RDBM table column locations as claimed.

Next, the Examiner suggests that Chang describes “*removing at least some of said data values from said XML document and storing the remainder of said XML document in said database as an XML skeleton which defines the structure of said XML document,*” citing Figures 8-12; col. 14, lines 19-50; col. 15, lines 24+; and col. 16, lines 56+ . These passages describe storing an XML document’s DTD which defines its structure and then using the DTD structure definition when performing indexed SQL structure queries. None of these passages suggests the claimed step of removing data values from an XML document and storing the remainder as an XML skeleton.

It is accordingly submitted that, contrary to the Examiner’s suggestion, Chang teaches neither the storage of element data in relational table column locations as claimed, nor does Chang teach removing data from an XML document and storing the remainder as a skeleton that defines the structure of the document.

The Examiner concedes that “Chang does not specifically disclose reconstructing said XML document by merging the data content of said specified rows with said XML skeleton.” That’s understandable since, as pointed out above, Chang stores neither the data content of the elements in tables, nor stores an XML skeleton, so neither part would be available to perform such a merger.

The secondary reference, Chau discloses a technique for automatically generating one or more XML documents from a relational database by mapping relational data to an XML structure. But there is nothing in the principle Chang reference that suggests the need for such a function, and accordingly nothing to suggest to one of ordinary skill any reason for combining the teachings as suggested by the Examiner.

Allowance of independent claims and its dependent claims 2-13 is accordingly requested.

**Independent claim 14 and its dependent claims 15-20**

Independent claim 14 recites parsing the XML document to identify element data and storing that data in specified column locations. As pointed out above in connection with claim 1, Chang does not suggest anything for performing these functions. Moreover, claim 14 recites means for performing relational operations on the data values stored in the data columns. The Chang system does not store data in that fashion and hence is unable to perform conventional relational operations on the element (column) data, but is instead forced to rely on mechanisms for searching the entire XML document stored as a single CLOB object using the XML extender operations.

As with claim 1, the Examiner concedes that "Chang does not specifically disclose reconstructing said XML document by merging the data content of said specified rows with said XML skeleton." But, as noted above, there is nothing in the Chang reference that would suggest any reason to merge data from a relational table into an XML document.

Allowance of independent claim 14 and its dependent claims 15-20 is accordingly requested.

**Dependent claims 2-7 and 10-44**

The Examiner asserts that Chang discloses that the subject matter set forth in claim 2, that the *"data value stored in each of said specified columns is obtained from a leaf element of said XML document which contains no sub element"* and is described in Figure 12 and at col. 16, lines 56-67 to col. 17-21. However, the cited passages describe how Chang uses the combination of data and structure index lists to speed searching, and nowhere suggests that element data is stored in table columns, let alone that the data placed in the columns is obtained from leaf elements in the XML document. Allowance of dependent claim 2, and its dependent claims 3-7 and 10-14 is accordingly requested for this additional reason.

**Dependent claims 3-7 and 10-14**

The Examiner suggests that Chang discloses that the subject matter set forth in claim 3, i.e., that the data values stored in specified rows of a table is obtained from an XML element which contains one or more leaf elements, with the data values in the leaf elements being stored

in columns each specified row. The Examiner Chang's Figure 12; col. 16, lines 56-67 to col. 17-21; and the Summary of the Chang invention. However, as noted above, nothing in Chang suggests that element data from an XML data is placed in specified columns and rows of a relational table in any fashion; instead, Chang teaches placing the entire XML document in a single column location. Allowance of claims 3-7 is accordingly requested for these additional reasons.

#### **Dependent claim 8**

The Examiner asserts that Chang discloses step of designating one or more XML elements as static elements and for identifying these static elements during said step of removing at least some of said data values to prevent removal of the data values in the static elements so that this static data is retained in said XML skeleton. The Examiner cites col. 14, line 19+ describing the enablement of an XML Index; and specifically col. 16, lines 54+ , suggesting that tag values are static elements which are retained in the structure index. First, it is again pointed out, as it noted above with respect to claim 1, that Chang does not teach removing data values from an XML document and storing the remainder as an XML skeleton that defines its structure. The Chang indexing method cited by the Examiner does no such thing. It creates data and structure index files by creating lists of tag and data value positions. It does not remove element data from an XML document to create a structure skeleton, and there is nothing in the cited passages that suggests that any kind of data removal, let alone selective removal of only the non-static element data. Allowance of claim 8 is accordingly requested for this additional reason.

#### **Dependent claims 9 and 15-17.**

The Examiner concedes that Chang does not specifically disclose the step of performing a relational database operation to modify the data value stored in at least one of the column locations such that the step of reconstructing an XML document produces a modified XML document. The Examiner suggests, however, that it would have been obvious at the time the invention was made to a person having ordinary skill in the art to modify the teaching of Chang with the teachings of Chau to include a method of creating triggers for insert, update, and delete data in an XML column before reconstructing an XML document from the data. It is submitted, however, as pointed out above with respect to claim 1, that Chang does not disclose placing

element data in columns but instead places the entire XML document in a single column. Hence, there is no need to reconstruct anything, and there would be no way to modify element data by modifying column data, because Chang doesn't store element data in columns. There is nothing in either or the cited references to suggest such a combination, and the rejection of claim 9 should be withdrawn.

#### **Dependent claims 16-17**

The Examiner suggests that Chang discloses relational operations that include queries which selectively retrieve and perform designated operations with respect to "said data values." However, the data values referred to are the element data values which are extracted from individual XML elements and placed in column locations. Chang's relational operations on element values are performed using the XML extender functions that operate on the entire XML document. Importantly, Chang's system cannot perform relational operations on individual XML element values stored in table columns as claimed because the element values are not separately stored in Chang's table columns.

#### **Dependent claims 19-20**

The Examiner suggests that Chang discloses means for removing one or more of the element data values from an XML document and for storing the remainder of the XML document as an XML skeleton. As noted above with respect to claim 1, Chang does not describe removing data from an XML document to produce a skeleton as claimed. The Examiner suggest that Chang [sic.] describes "*means for reconstructing said XML document comprises means for re-inserting said data values from said relational tables into said XML skeleton and for generating the merged combination of said skeleton and said data values as an XML document*", citing Chau's teaching of a mechanism for inserting relational table data into an output XML document.; however, as noted above, there is no reason to modify Chang's system to place relational column data into an XML document, because Chang never decomposes the XML document and doesn't need to reconstruct it.

**Conclusion**

The cited references, whether considered singly or in combination, fail to disclose of suggest the subject matter set forth in claims 1-20, and allowance of all claims is accordingly requested.

Respectfully submitted,



Charles G. Call, Reg. 20,406

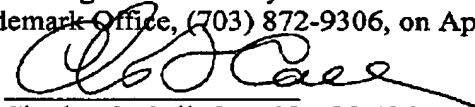
Dated: April 28, 2004

**Certificate of Transmission under 37 CFR 1.8**

I hereby certify that this *Amendment* is being transmitted by facsimile to the central facsimile number of the U.S. Patent and Trademark Office, (703) 872-9306, on April 28, 2004.

Dated: April 28, 2004

Signature



Charles G. Call, Reg. No. 20,406

USPTO Customer No. 021253

68 Horse Pond Road

West Yarmouth, MA 02673

Ph. (508) 778-2630 Fax (508) 629-6540

call@patentsoft.com